# LAPORAN PRAKTIKUM DATA STRUCTURE

# SIMPLE SORTING

**Dosen Pengampu:**

**H. Fatchurrochman,M.Kom**

**Asisten Praktikum:**

**Fillah Anjany 230605110033**


**Oleh :**

**Muhammad Alif Mujaddid** **240605110082**

**Jurusan Teknik Informatika Fakultas Sains dan Teknologi**

**UIN Maulana Malik Ibrahim Malang**

**2025**

1. **Sorting Object**

   Pada praktikum 1 (Arrays), Anda telah mengiimplementasikan storing object "mahasiswa". Implementasikan algoritma sorting untuk sorting object "Mahasiswa" berdasarkan NIM. Yaitu dengan cara menambahkan method BubbleSort( ), SelectionSort( ), dan InsertionSort( ) sebagaimana yang dituliskan pada tugas pendahuluan modul ini (dengan sedikit penyesuaian) pada class DataArray (listing nomer 3 parktikum 1). Method tersebut dipanggil pada class DataArrayApp.

   Keterangan:

   Praktikan dengan NIM genap mengerjakan **BubbleSort( ) dan SelectionSort( )**

2. **Lexicographical Comparisons**

   NIM pada object mahasiswa adalah variable bertipe long. Anda dapat membandingkan item dengan tipe long dengan menggunakan operator equality relational. Pada sorting object, kita perlu mengetahui cara membandingkan item dengan tipe String (object). Misal, adakalanya pencarian record mahasiswa dapat menggunakan field "nama" sebagai key. Untuk kondisi ini, kita dapat menggunakan method compareTo() yang ada pada class String. Method compareTo() membandingkan dua string secara leksikograf (alfabetis). Perbandingan ini didasarkan pada nilai Unicode dari masing-masing karakter dalam String. Deretan karakter objek String ini dibandingkan secara leksikograf dengan deretan karakter string argumen. Value yang dikembalikan dalam pemanggilan method ini adalah nilai 0 jika objek string sama dengan string argumen; nilai kurang dari 0 jika objek string secara leksikografis kurang dari string argumen; dan nilai lebih besar dari 0 jika objek string secara leksikograf lebih besar dari string argumen. (lihat Tabel 2.1)

   Tabel 2.1 Operasi pada method `compareTo()`

| `StringObjek.compareTo(StringArgumen)` | **Return value** |
|---|---|
| `StringArgumen` < `StringObjek` | < 0 |
| `StringArgumen` equal `StringObjek` | 0 |
| `StringArgumen` > `StringObjek` | > 0 |

   Variasi lain dari method compareTo() adalah compareToIgnoreCase(). Method ini memiliki fungsi yang sama dengan method compareTo() namun dengan mengabaikan besar/kecilnya huruf. Gunakan method compareTo() atau compareToIgnoreCase()pada tahap perbandingan item untuk melakukan sorting object "mahasiswa" berdasarkan field "nama"!

   Keterangan: - Praktikan dengan NIM genap mengerjakan **InsertionSortbyName( )**

**Source Code :**

```java
public void bubblesort() {
    int i;
    Mahasiswa temp;
    for (i = nElemen - 1; i > 0; i--) {
        for (int j = 0; j < i; j++) {
            if (mhs[j].getNim() > mhs[j + 1].getNim()) {
                temp = mhs[j];
                mhs[j] = mhs[j + 1];
                mhs[j + 1] = temp;
            }
        }
    }
}
```

```java
public void selectionsort() {
    int i, j, min;
    Mahasiswa temp;
    for (i = 0; i < nElemen - 1; i++) {
        min = i;
        for (j = i + 1; j < nElemen; j++) {
            if (mhs[j].getNim() < mhs[min].getNim()) {
                min = j;
            }
        }
        temp = mhs[i];
        mhs[i] = mhs[min];
        mhs[min] = temp;
    }
}
```

```java
public void sortByName() {
    for (int i = 1; i < nElemen; i++) {
        Mahasiswa key = mhs[i];
        int j = i - 1;
        while (j >= 0 && mhs[j].getNama().compareTo(key.getNama()) > 0) {
            mhs[j + 1] = mhs[j];
            j--;
        }
        mhs[j + 1] = key;
    }
}
```

```java
public static void main(String[] args) {
    DataArray arr;
    arr = new DataArray(maxSize);
    arr.insert(nim:16650210, nama:"ahmad", asal:"sidoarjo");
    arr.insert(nim:16650200, nama:"jundi", asal:"malang");
    arr.insert(nim:16650220, nama:"ismail", asal:"Banyuwangi");
    arr.insert(nim:16650250, nama:"rais", asal:"ambon");
    arr.insert(nim:16650230, nama:"sofi", asal:"semarang");
    arr.insert(nim:16650240, nama:"dinda", asal:"bandung");
    arr.insert(nim:16650260, nama:"helmi", asal:"madura");
    arr.insert(nim:16650270, nama:"agung", asal:"madiun");
    arr.insert(nim:16650280, nama:"arina", asal:"malang");
    System.out.println(x:"sebelum sort :");
    arr.display();
    System.out.println(x:"------------------------------------------------------------");
    System.out.println(x:"buble sort :");
    arr.bubblesort();
    arr.display();
    System.out.println(x:"------------------------------------------------------------");
    System.out.println(x:"Selecti sort :");
    arr.selectionsort();
    arr.display();
    System.out.println(x:"------------------------------------------------------------");
    System.out.println(x:"insertion sort by name :");
    arr.sortByName();
    arr.display();
```

Output :

```
sebelum sort :
        nim: 16650210,  nama: ahmad,    asal: sidoarjo,
        nim: 16650200,  nama: jundi,    asal: malang,
        nim: 16650220,  nama: ismail,   asal: Banyuwangi,
        nim: 16650250,  nama: rais,     asal: ambon,
        nim: 16650230,  nama: sofi,     asal: semarang,
        nim: 16650240,  nama: dinda,    asal: bandung,
        nim: 16650260,  nama: helmi,    asal: madura,
        nim: 16650270,  nama: agung,    asal: madiun,
        nim: 16650280,  nama: arina,    asal: malang,
============================================================
buble sort :
        nim: 16650200,  nama: jundi,    asal: malang,
        nim: 16650210,  nama: ahmad,    asal: sidoarjo,
        nim: 16650220,  nama: ismail,   asal: Banyuwangi,
        nim: 16650230,  nama: sofi,     asal: semarang,
        nim: 16650240,  nama: dinda,    asal: bandung,
        nim: 16650250,  nama: rais,     asal: ambon,
        nim: 16650260,  nama: helmi,    asal: madura,
        nim: 16650270,  nama: agung,    asal: madiun,
        nim: 16650280,  nama: arina,    asal: malang,
============================================================
selecti sort :
        nim: 16650200,  nama: jundi,    asal: malang,
        nim: 16650210,  nama: ahmad,    asal: sidoarjo,
        nim: 16650220,  nama: ismail,   asal: Banyuwangi,
        nim: 16650230,  nama: sofi,     asal: semarang,
        nim: 16650240,  nama: dinda,    asal: bandung,
        nim: 16650250,  nama: rais,     asal: ambon,
        nim: 16650260,  nama: helmi,    asal: madura,
        nim: 16650270,  nama: agung,    asal: madiun,
        nim: 16650280,  nama: arina,    asal: malang,
============================================================
```

```
============================================================
insertion sort by name :
        nim: 16650270,  nama: agung,    asal: madiun,
        nim: 16650210,  nama: ahmad,    asal: sidoarjo,
        nim: 16650280,  nama: arina,    asal: malang,
        nim: 16650240,  nama: dinda,    asal: bandung,
        nim: 16650260,  nama: helmi,    asal: madura,
        nim: 16650220,  nama: ismail,   asal: Banyuwangi,
        nim: 16650200,  nama: jundi,    asal: malang,
        nim: 16650250,  nama: rais,     asal: ambon,
        nim: 16650230,  nama: sofi,     asal: semarang,
```

C. KESIMPULAN

Kesimpulan yang diperoleh dari pembahasan praktikum kali ini adalah:

1. Tentang perbandingan algoritma sorting bubble, selection, dan insertion:

Berdasarkan percobaan perbandingan algoritma sorting Bubble Sort, Selection Sort, dan Insertion Sort pada data Mahasiswa, diperoleh kesimpulan sebagai berikut:

1. Bubble Sort bekerja dengan cara menukar elemen secara berulang dengan membandingkan pasangan berurutan, sehingga memiliki kompleksitas waktu $O(n^2)$ dan relatif lambat untuk data yang besar.
2. Selection Sort mencari elemen terkecil pada setiap iterasi dan menempatkannya di posisi yang benar. Kompleksitas waktunya juga $O(n^2)$, namun jumlah pertukaran (swap) lebih sedikit dibandingkan Bubble Sort.
3. Insertion Sort bekerja dengan cara menyisipkan elemen ke dalam posisi yang sesuai pada bagian array yang sudah terurut. Algoritma ini juga memiliki

kompleksitas $O(n^2)$ pada kasus terburuk, tetapi lebih efisien dibandingkan Bubble Sort dan Selection Sort pada data yang hampir terurut, bahkan bisa mendekati $O(n)$.

2.Tentang sorting object :

Pada praktikum ini, kami belajar tentang sorting object, yaitu mengurutkan data yang berbentuk objek, bukan hanya angka biasa. Objek yang digunakan adalah Mahasiswa yang punya atribut nim, nama, dan asal. Dalam program, kami mencoba beberapa algoritma sorting seperti Bubble Sort, Selection Sort, dan Insertion Sort. Untuk sorting berdasarkan nim, perbandingan bisa langsung dilakukan karena nilainya berupa angka. Sedangkan kalau sorting berdasarkan nama, perbandingan dilakukan menggunakan method compareTo() dari Java supaya bisa menentukan urutan sesuai alfabet. Dengan cara ini, data mahasiswa bisa diurutkan sesuai kebutuhan, misalnya dari nim terkecil ke terbesar, atau dari nama A sampai Z. Dari percobaan, terlihat bahwa walaupun algoritma berbeda, hasil akhirnya sama-sama bisa mengurutkan data mahasiswa dengan benar, hanya saja ada perbedaan dalam cara kerja dan kecepatan algoritmanya.